Convolutional Neural Networks to Enhance Coded Speech

Ziyue Zhao, Huijun Liu, and Tim Fingscheidt, *Senior Member, IEEE*

***Abstract*—Enhancing coded speech suffering from far-end acoustic background noise, quantization noise, and potentially transmission errors is a challenging task. In this paper, we propose two postprocessing approaches applying convolutional neural networks either in the time domain or the cepstral domain to enhance the coded speech without any modification of the codecs. The time-domain approach follows an end-to-end fashion, whereas the cepstral domain approach uses analysis–synthesis with cepstral domain features. The proposed postprocessors in both domains are evaluated for various narrowband and wideband speech codecs in a wide range of conditions. The proposed postprocessor improves perceptual evaluation of speech quality by up to 0.25 mean opinion score listening quality objective points for G.711, 0.30 points for G.726, 0.82 points for G.722, and 0.26 points for adaptive multi-rate wideband codec. In a subjective comparison category rating listening test, the proposed postprocessor on G.711-coded speech exceeds the speech quality of an ITU-T-standardized postfilter by 0.36 CMOS points, and obtains a clear preference of 1.77 CMOS points compared to legacy G.711, *even better than uncoded speech with statistical significance*. The source code for the cepstral domain approach to enhance G.711-coded speech is made available.[1]**

***Index Terms*—Convolutional neural networks, speech codecs, speech enhancement.**

I. Introduction

SPEECH signals being subject to speech encoding, transmission, and decoding are often called transcoded speech, or simply: coded speech. Coded speech often suffers from far-end acoustic background noise, quantization noise, and potentially transmission errors. To enhance the quality of coded speech, postprocessing methods, operating just after speech decoding can be advantageously employed.

To combat quantization noise at the receiver, a postfilter based on classical Wiener theory of optimal filtering has been standardized for the logarithmic pulse code modulation (PCM) G.711 codec [1]. It is part of the G.711 audio quality enhancement toolbox [2], described in detail in the appendix of G.711.1 [3], a wideband extension of G.711. This postfilter uses *a priori* information on the A- or μ-law properties to estimate the quantization noise power spectral density (PSD), assuming the quantization noise to be spectrally white [4], [5]. Then, a Wiener filter is derived by the estimation of the *a priori* signal-to-noise-ratio (SNR) based on a two-step noise reduction approach [6]. After the filtering process, a limitation of distortions is performed to control the waveform difference between the original signal and the postprocessed coded signal.

However, as the bitrates go down for most of the modern codecs, it becomes more difficult for the classical Wiener filter to effectively suppress the quantization noise, while maintaining the speech perceptually undistorted, since the SNR drops and more importantly, only the mean squared error (MSE) is minimized in the Wiener filter [7]. Therefore, some perceptually-based postfilters have been proposed to reduce the perceptual degradation caused by low bitrate codecs. Formant enhancement postfilters [8], [9] emphasize the peaks of the spectral envelope while further suppressing the valleys to reduce the impact of quantization noise in coded speech, since the formants are perceptually more important than the spectral valleys. This type of postfilter typically consists of three parts [9]: The core short-term postfilter to enhance the formants, a tilt correction filter to compensate the low-pass tilt caused by the core postfilter, and an adaptive gain control to compensate the gain misadjustment caused by parts one and two.

In addition to modifying the spectral envelope of the speech signal, the spectral fine structure of voiced speech is improved by a pitch enhancement postfilter, aiming to emphasize the harmonic peaks and attenuate the gaps between the harmonics [9]. In practice, this long-term postfilter is always applied to low frequencies, where harmonic peaks are more prominent, which actually forms a bass postfilter [10]. This bass postfilter and the formant enhancement postfilter are used either together or separately in the decoders of some standard codecs, e.g., in adaptive multi-rate (AMR) [11], wideband AMR (AMR-WB) [12] and enhanced voice services (EVS) [13].

For speech codecs using the so-called algebraic code-excited linear prediction (ACELP) codebooks, e.g., AMR and AMR-WB, an anti-sparseness postprocessing procedure is applied, aiming to suppress the perceptual artifacts caused by the sparseness of the algebraic fixed codebook vectors with only a few non-zero pulses per subframe, especially in low bitrate modes [11], [12]. A modification of the fixed codebook vector is adaptively selected based on the quantized adaptive codebook gain [14].

In an attempt to combat quantization noise, it has been shown that if residual correlation exists in coded signals [15]–[17] or more specifically, coded speech [18], a time-variant receiver-sided codebook or a shallow neural network can provide some gains in a system-compatible fashion.

Apart from the aforementioned quantization noise, also far-end acoustic background noise can degrade the quality and intelligibility of coded speech. In most cases, noise reduction

Manuscript received April 16, 2018; revised August 22, 2018, October 17, 2018, and November 29, 2018; accepted December 12, 2018. Date of publication December 17, 2018; date of current version February 15, 2019. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Xiaodong Cui. *(Corresponding author: Tim Fingscheidt.)*

The authors are with the Institute for Communications Technology, Technische Universität Braunschweig, Braunschweig 38106, Germany (e-mail: ziyue.zhao@tu-bs.de; h.liu@tu-bs.de; t.fingscheidt@tu-bs.de).

Digital Object Identifier 10.1109/TASLP.2018.2887337

[1]https://github.com/ifnspaml/Enhancement-Coded-Speech

2329-9290 © 2018 IEEE. Translations and content mining are permitted for academic research only. Personal use is also permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Fig. 1. General flow chart of postprocessing for enhancement of coded speech.

approaches are conducted as a transmitter-sided preprocessing step to suppress the background noise before the speech signal is coded and transmitted [19]. However, since the noise usually cannot be entirely suppressed and therefore speech with some residual noise is coded and transmitted to the receiver side, one can aim to further reduce the noise of the coded speech in the postprocessing procedure. To accomplish this, a modified post-filter has been proposed for speech quality enhancement, where the parameters corresponding to the formant and pitch emphasis are adaptively updated based on the statistics of the background noise [20]. Furthermore, in adverse noise conditions, also post-filtering methods to improve the speech intelligibility have been studied [21]. Additionally, a kind of postprocessing to enhance the coded speech in transmitter-sided noisy environments by restoring the distorted background noise while masking main coding artifacts for low bitrate speech coding is proposed and standardized in EVS as comfort noise addition [13]. An artificial comfort noise is generated and added to the coded speech signal after the level and the spectral shape of the background noise are estimated [22].

Recently, speech enhancement based on neural networks has been intensively studied [23]–[37]. Deep neural networks (DNNs) are used as a classification method to estimate the ideal binary mask [23] or smoothed ideal ratio mask [24] for noise reduction. Also, some regression approaches based on DNNs to learn a mapping function from noisy to clean speech features have been proposed [25], [26]. Furthermore, a deep denoising autoencoder is applied for noise reduction, with either both clean pairs [27] or noisy and clean pairs [28] as inputs and targets to train the autoencoder. Besides, recurrent neural networks (RNNs) are used for speech enhancement, e.g., a recurrent denoising autoencoder for robust automatic speech recognition (ASR) [29] and long short-term memory (LSTM) structure for noise reduction [30], [31].

In addition to the DNNs and RNNs, convolutional neural networks (CNNs) are achieving increasing attention for the speech enhancement task [32]–[37]. The CNNs are trained to learn a mapping between the noisy speech features and the clean speech features, e.g., log-power spectrum [32]–[34] or complex spectrogram [35], or a mapping directly between the noisy raw speech waveform and clean raw speech waveform [36], [37]. The convolutional layers in the CNNs have the property of local filtering, i.e., the input features share the same network weights, resulting in translation invariance for the output of the network, which is a desired property for the modeling of speech [38]. This local filtering property makes the CNNs have the ability to characterize local information of the speech signal, which clearly provides benefits for the task of speech enhancement. It is also because of this property that the number of the trainable weights is reduced in a large scale compared to DNNs and RNNs with fully-connected structures, making it more efficient to train the network [32].

In this work, we use CNNs to enhance *coded* speech, so that this operation can be seen as a postprocessor after speech decoding (or anywhere later in the transmission chain) aiming at improving speech quality at the far-end, which is different to the aforementioned noise reduction approaches. Fig. 1 shows the general flow chart of postprocessing for coded speech. Motivated by the successful application of CNNs to the image super-resolution problem in computer vision [39]–[42], aiming at restoring the missing information from the low-resolution image, we propose to use similar convolutional network structures to restore improved speech from speech being subject to encoding and decoding. In terms of the topology, we adopt the deep convolutional encoder-decoder network topology [40], which is a symmetric structure with multiple layers of convolution and deconvolution [43], [44], in order to firstly preserve the major information of the input features and meanwhile reduce the corruption and then recover the details of the features [40], [41]. Furthermore, skip-layer connections are added symmetrically between the convolution and deconvolution layers to form a residual network for an effective training [42], [45].

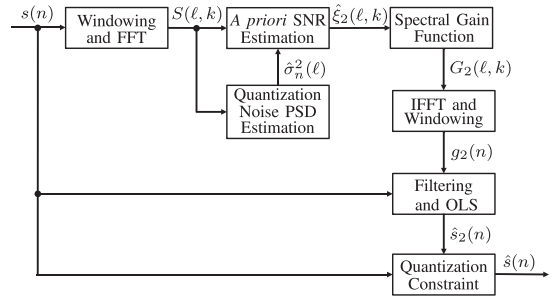


Fig. 2. The postprocessing flow chart of the G.711 Amendment 2: New Appendix III audio quality enhancement toolbox (see [2]).

The contribution of this work is threefold: First, based on the CNN topology, we propose two different postprocessing approaches in the time domain and the cepstral domain to restore the speech either in an end-to-end fashion or in an analysis-synthesis fashion with cepstral domain features. To our knowledge, it is the first time that deep learning methods are used to enhance coded speech. Second, we show by objective and subjective listening quality assessment that both proposed approaches show superior performance compared to the state of the art G.711 postprocessing. Finally, both proposed approaches are system-compatible for different kinds of codecs without any modification of the encoder or decoder. The simulation results in clean and noisy speech conditions, tandeming, and frame loss conditions show their effectiveness for some widely used speech codecs in narrowband and wideband.

The article is structured as follows: In Section II we briefly sketch state of the art G.711 postprocessing, which serves as a baseline method in the evaluation part. Next, we describe the proposed CNN postprocessing approaches in both time domain and cepstral domain in Section III. Subsequently, the experimental setup and the instrumental metrics for speech quality evaluation are explained in Section IV. Then, in Section V, we present the evaluation results and discussion. Finally, we conclude our work in Section VI.

II. The G.711 Postprocessing Baseline

In Fig. 2 the G.711 postprocessing aiming at attenuation of quantization noise is depicted. It has originally been proposed in [4] and standardized in [2], basically following the classical framework of noise reduction, comprising: quantization noise power spectral density (PSD) estimation, *a priori* SNR estimation, spectral weighting rule using the Wiener filter, and finally a quantization constraint. In the following subsections,

this G.711 postprocessing is briefly reviewed as our baseline for enhancement of G.711-coded speech.

A. Quantization Noise PSD Estimation

At first the coded speech $s(n)$ is subject to a periodic Hann window and then being transformed to the frequency domain representation $S(\ell, k)$ via the fast Fourier transform (FFT), with ℓ being the frame index and k being the frequency bin index. Since the quantization noise of G.711 is assumed to be spectrally white, the estimate of the quantization noise variance $\sigma_n^2(\ell)$ is sufficient for the quantization noise PSD estimation. To achieve this, an estimate of the (uncoded) source speech signal variance $\hat{\sigma}_{\tilde{s}}^2(\ell)$ is needed first, and subsequently an estimate of the load factor, defined as $\hat{\Gamma}(\ell) = 1/\hat{\sigma}_{\tilde{s}}(\ell)$ denoting how the signal exploits the quantizer dynamic, is achieved. Interestingly, the estimate of the uncoded signal variance $\hat{\sigma}_{\tilde{s}}^2(\ell)$ is actually obtained by estimating the coded signal variance $\hat{\sigma}_s^2(\ell)$, assuming the variance of the quantization noise to be very low compared to the uncoded signal most of the time [4]:

$$\hat{\sigma}_{\tilde{s}}^2(\ell) \approx \hat{\sigma}_s^2(\ell) = \frac{1}{|\mathcal{N}_\ell|} \sum_{n \in \mathcal{N}_\ell} s^2(n). \tag{1}$$

The set $\mathcal{N}_\ell$ contains all sample indices n belonging to frame ℓ and $|\mathcal{N}_\ell|$ is the number of samples in the frame. Then the signal-to-quantization-noise ratio is obtained according to the estimated load factor $\hat{\Gamma}(\ell)$ and the A- or μ-law function. Finally, the estimate of the (spectrally white) quantization noise variance $\hat{\sigma}_n^2(\ell)$ is obtained.

B. A Priori SNR Estimation and Wiener Filtering

After estimation of the noise PSD, the *a priori* SNR is obtained by a two-step noise reduction technique [6] and subsequently the Wiener filter results. In order to estimate the *a priori* SNR, the *a posteriori* SNR is computed first as

$$\gamma(\ell, k) = \frac{|S(\ell, k)|^2}{\hat{\sigma}_n^2(\ell)}. \tag{2}$$

Then, the first-step spectral gain function $G_1(\ell, k)$ from the Wiener filter can be expressed as

$$G_1(\ell, k) = \frac{\hat{\xi}_1(\ell, k)}{1 + \hat{\xi}_1(\ell, k)}, \tag{3}$$

where the first-step *a priori* SNR estimate $\hat{\xi}_1(\ell, k)$ from the decision-direction approach [46] is

$$\hat{\xi}_1(\ell, k) = \beta \frac{|\hat{S}_1(\ell-1, k)|^2}{\hat{\sigma}_n^2(\ell-1)} + (1-\beta)\max\big(\gamma(\ell, k) - 1, 0\big), \tag{4}$$

with $\hat{S}_1(\ell-1, k) = G_1(\ell-1, k)S(\ell-1, k)$ and β being a weighting factor. In the second step, an updated spectral gain function is computed as

$$G_2(\ell, k) = \max\left(\frac{\hat{\xi}_2(\ell, k)}{1 + \hat{\xi}_2(\ell, k)}, G_{\min}\right), \tag{5}$$

where $G_{\min}$ is the lower limit to avoid over-attenuation and

$$\hat{\xi}_2(\ell, k) = \frac{|G_1(\ell, k)S(\ell, k)|^2}{\hat{\sigma}_n^2(\ell)} \tag{6}$$

is the updated *a priori* SNR estimate. Finally, a causal filter impulse response $g_2(n)$ is obtained from this updated spectral gain function (5) by inverse FFT (IFFT) and imposing a linear phase, and the coded speech $s(n)$ is time-domain-filtered and the overlap and save (OLS) method provides the enhanced speech $\hat{s}_2(n)$. Note that due to its frame structure, the G.711 postfilter baseline has an algorithmic delay of 2 ms.

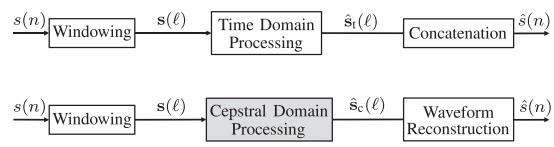


Fig. 3. **CNN-based postprocessing** for **time domain** approach (upper) and **cepstral domain** approach (lower). More details of the cepstral domain processing can be found in Figs. 4, 5, and 6.

C. Quantization Constraint

In order to avoid extra distortion introduced by the above postprocessing, finally a limitation of potential distortions is performed. Since the quantization interval of each coded speech sample $s(n)$ is known, this idea is to limit the postprocessed samples $\hat{s}(n)$ to lie within the respective interval. If an outlier sample (outside the certain quantization interval) is detected, the constraint will replace it by the closest decision boundary of this respective quantization interval. After application of this constraint, the final postprocessed speech $\hat{s}(n)$ is obtained.

III. Convolutional Neural Network (CNN) Postprocessing

In this section, we present the proposed CNN-based postprocessing for coded speech alternatively in the time domain and in the cepstral domain. Fig. 3 depicts the high-level block diagram. At first, for both approaches, the coded speech $s(n)$ is assembled to frames $\mathbf{s}(\ell)$, applying a window function. Then, the frame is processed either in the time domain resulting in $\hat{\mathbf{s}}_{\mathrm{t}}(\ell)$, or in the cepstral domain resulting in $\hat{\mathbf{s}}_{\mathrm{c}}(\ell)$. Finally, the enhanced speech $\hat{s}(n)$ is obtained via either a direct concatenation of the processed frames $\hat{\mathbf{s}}_{\mathrm{t}}(\ell)$ for the time domain approach, or some waveform reconstruction of the processed frames $\hat{\mathbf{s}}_{\mathrm{c}}(\ell)$ for the cepstral domain approach, as outlined in the following.

A. Time Domain Approach: Processing

For the time domain approach, we choose a quite straightforward framework structure (i.e., windowing and waveform reconstruction) which fits to most speech decoders: a 10 ms rectangular window without overlapping. The windowed frame $\mathbf{s}(\ell)$ then serves directly as the input of the CNN with the target being $\tilde{\mathbf{s}}(\ell)$, which is the noise-free undistorted (uncoded) windowed speech frame. Details of the CNN topology will be presented in Section V-A. After CNN processing, the enhanced frame $\hat{\mathbf{s}}_{\mathrm{t}}(\ell)$ is directly concatenated to reconstruct the waveform $\hat{s}(n)$. The motivation of this end-to-end time domain approach is to learn a mapping from the coded speech frame to the undistorted speech frame via the CNN, exploiting the temporal redundancy in terms of speech signal correlation in the decoder, to directly enhance the waveform of the coded speech. Beyond framing, no additional algorithmic delay is incurred. Note that this allows effectively latency-free postfiltering if the frame size matches the frame size of the speech decoder or if it matches the voice-over-IP packet size.

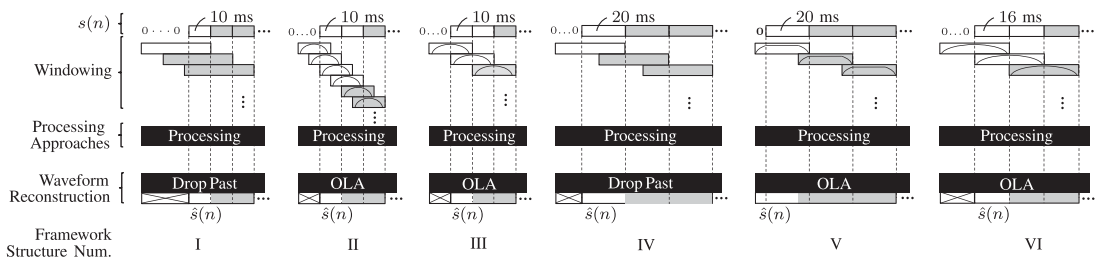


Fig. 4. **Framework structures** for windowing, (**cepstral domain**) processing, and waveform reconstruction. In the upper part of the figure, all signal portions necessary to be available for computing the first frame ℓ of $\hat{s}(n)$ are marked as white boxes [***] , as is the current output frame $\hat{s}(n), n \in \mathcal{N}_\ell$, in the bottom part of the figure. OLA stands for overlap-add of all upper-part white windowing boxes for current frame ℓ.

TABLE I
DETAILED SETTINGS OF THE FRAMEWORK STRUCTURES FOR THE **CEPSTRAL DOMAIN** APPROACH

	Framework Structure					
	I	II	III	IV	V	VI
Window length N_w [ms]	32	15	20	32	25	32
Processing length [ms]	32	16	32	32	32	32
Processing shift N_s [ms]	10	5	10	20	20	16
Output overlap ratio	0	66.7%	50%	0	20%	50%
Additional delay [ms]	0	10	10	0	5	16

B. Cepstral Domain Approach: Framework Structures

This subsection presents the various framework structures for the cepstral domain approach, shown in Fig. 4. On the one hand, since FFT and discrete cosine transform (DCT) are performed in the cepstral domain approach to obtain the cepstral coefficients (explained in detail in Section III-C), an appropriate frame length and overlapping setting are important. On the other hand, since the postprocessor follows the speech decoder, the frame lengths of typical decoders are also taken into consideration to design the framework structures. As a result, we investigate six framework structures to offer broad selections for various possible application scenarios. These structures can be divided into three groups: structures I, II and III are designed for codecs with 10 ms frames, IV and V are designed for codecs with 20 ms frames, while structure VI is for delay-insensitive off-line usage with 16 ms frames, one frame lookahead, and 50% overlap.

First of all, windowing of the coded speech $s(n)$ is implemented to form frames for processing, which can be denoted as

$$\mathbf{s}(\ell) = \big[s\big((\ell-1)N_\mathrm{s}\big), \ldots, s\big((\ell-1)N_\mathrm{s} + N_\mathrm{w} - 1\big)\big] \circ \mathbf{w}, \quad (7)$$

where N_s is the frame shift, N_w is the length of window function, $\mathbf{w}$ is the window function vector, and $\circ$ denotes the sample-wise multiplication. As shown in Fig. 4, all six frameworks require a few initial zeros to be padded to the coded input speech. The detailed settings of the framework structures are listed in Table I. It is worth noting that if the processing length is longer that the window length, a zero-padding is performed also after windowing.

After processing of the windowed frames, the speech waveform needs to be reconstructed, which is also illustrated in Fig. 4. In structure I and structure IV, only the latest samples of the processed frame are kept and the other samples are dropped, which means that beyond framing (10 ms and 20 ms, respectively) no additional algorithmic delay coccus. If used in conjunction with speech decoders operating with this frame size, or if used in conjunction with, e.g., G.711, G.726, or G.722, assembled to 10 ms voice-over IP packets, the entire postprocessing is effectively free of algorithmic delay (as is the case in the time domain approach, cf. Section III-A). In structures II, III and VI, since periodic Hann windows are employed, the processed frames overlap and need to be added after time alignment. As a result, additional algorithmic delay is introduced for each of these three structures. Structure V aims at low complexity by using a flat-top periodic Hann window with low overlap ratio. In this structure, the output signal will be delayed by only 5 ms, i.e., the output starts with 5 ms of zeros.

C. Cepstral Domain Approach: Processing

As we have learnt from the aforementioned formant postfilters, an emphasis of the spectral envelope peaks can reduce the impact of the coding distortion. By using cepstral domain envelope features, the dimension of the input vector to the CNN will be largely reduced compared to the time domain approach, which makes the CNN able to concentrate on the more perceptually relevant information, i.e., the formant structure.

Our cepstral domain approach uses a CNN to restore the cepstral coefficients responsible for the spectral envelope and then synthesizes the speech frame using the enhanced envelope cepstral coefficients, as well as the residual cepstral coefficients and the phase information, the two latter both being acquired from the coded speech frame. The whole processing structure is shown in Fig. 5.

At first, the windowed frame is transformed to the frequency domain as vector $\mathbf{S}(\ell)$ using the K-point FFT. Subsequently, the cepstrum (cepstral coefficients) is computed by applying the discrete cosine transform of type II (DCT-II) to the logarithmic magnitude spectrum, which can be expressed as

$$c(\ell, m) = \sum_{k \in \mathcal{K}} \log(|S(\ell, k)|) \cdot \cos\big(\pi m(k + 0.5)/K\big), \quad (8)$$

where $k \in \mathcal{K} = \{0, \cdots, K-1\}$ is the frequency bin index and $m \in \mathcal{M} = \{0, 1, \cdots, K-1\}$ is the index of cepstral coefficients. Then, the cepstrum is lowpass liftered (i.e., taking only the lower part of the cepstrum) to obtain the cepstral coefficients responsible for the spectral envelope, which is denoted as $c_\mathrm{env}(\ell, m)$ with $m \in \mathcal{M}_\mathrm{env}$. In this work, we regard the first 6.25% cep-

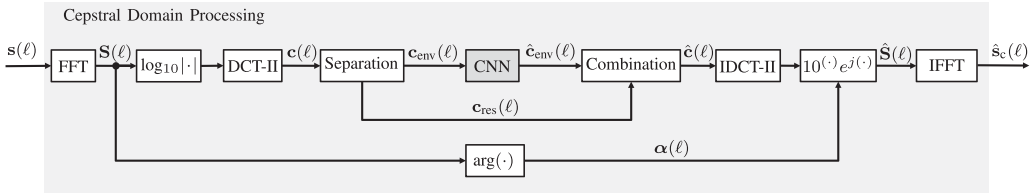


Fig. 5. **Processing structure of the cepstral domain approach**. The topology of the CNN block is identical in the time domain and the cepstral domain approach and is depicted in Fig. 6.

stral coefficients as the coefficients responsible for the spectral envelope[2], resulting in $|\mathcal{M}_{\text{env}}| = 6.25\% \cdot |\mathcal{M}|$.

This vector $\mathbf{c}_{\text{env}}(\ell)$ serves as the input to the CNN, which then provides the restored cepstral coefficients responsible for the spectral envelope $\hat{\mathbf{c}}_{\text{env}}(\ell)$. After that, the residual cepstral coefficients from the liftering, denoted as $c_{\text{res}}(\ell, m)$ with $m \in \mathcal{M}_{\text{res}}$, are concatenated to $\hat{\mathbf{c}}_{\text{env}}(\ell)$ to constitute the complete cepstral coefficient vector $\hat{\mathbf{c}}(\ell)$. Then the logarithmic magnitude of the processed spectrum $\hat{\mathbf{S}}(\ell)$ is calculated by inverse DCT-II (IDCT-II) as

$$\log\left|\hat{S}(\ell,k)\right| = \frac{1}{K}\left[\hat{c}(\ell,0) + 2\sum_{m=1}^{K-1}\hat{c}(\ell,m)\cdot\cos\left(\frac{\pi m(k+0.5)}{K}\right)\right]. \tag{9}$$

Finally, the elements of $\hat{\mathbf{S}}(\ell)$ are subsequently obtained by

$$\hat{S}(\ell,k) = \left|\hat{S}(\ell,k)\right| \exp(j \cdot \alpha(\ell,k)), \tag{10}$$

where $\boldsymbol{\alpha}(\ell)$ is the phase information from $\mathbf{S}(\ell)$. The processed frame $\hat{\mathbf{s}}_{\text{c}}(\ell)$ is obtained by performing the IFFT of $\hat{\mathbf{S}}(\ell)$.

D. Both Approaches: CNN Topology

The CNN topology, both in the time domain approach or in the cepstral domain approach, is a deep convolutional encoder-decoder network, which is shown in Fig. 6. This topology is motivated from [40] and three different kinds of layers are used in this topology which will be explained in the following.

The *convolutional layers* are defined by the number F or $2F$ of feature maps (filter kernels) and the kernel size ($a \times b$). The number of trainable weights, including the bias, of a convolutional layer denoted as, e.g., the first layer (Conv(F,$N \times 1$)), results in $F \times (N \times 1) + F$. It is worth noting that in each convolutional layer, the stride is 1 and zero-padding of the layer input is always performed to guarantee that the first dimension of the layer output is the same as that for the layer input. In *max pooling layers*, a 2×1 maximum filter is applied in a non-overlapping fashion, resulting in a 50% reduction of the layer input along the first dimension. On the contrary, the *upsampling layers* simply copy each element of the layer input into a 2×1 vector and stack these vectors just following the original order, which actually doubles the first dimension of the layer input.

[2]As we have $K = 512$ for narrowband speech, the $0.0625 \cdot K$-th $= 32$nd cepstral coefficient represents the frequency $1/(32 \times \frac{1}{16}\text{ ms}) = 500$ Hz (check (8) !). Using 500 Hz as cepstral lowpass liftering cutoff frequency, the fundamental frequency (F0) will be excluded in most cases. This is because the fundamental frequency can vary from 40 Hz for a very low-pitched male voice to 600 Hz for a very high-pitched female or child voice [47]. As a result, the pitch periodicity from speech is removed, while the information of the spectral envelope representing the formants is kept for further processing.

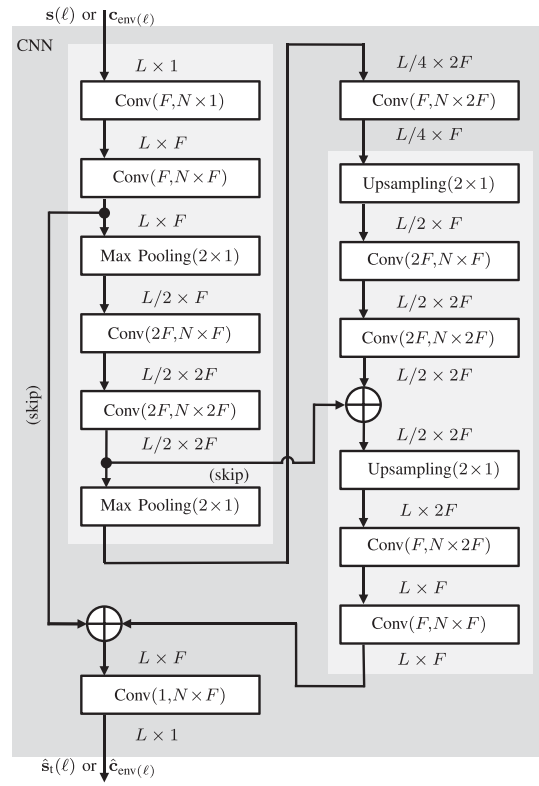


Fig. 6. Detailed view of the **CNN structure in both time domain** (L equals 10 ms of speech samples) and **cepstral domain** ($L = |\mathcal{M}_{\text{env}}|$). The operation Conv() stands for convolutional layers containing two parameters, which are the number of feature maps (filter kernels) F or $2F$, and the kernel size ($a \times b$). The max pooling and upsampling layers are described by the kernel size (2×1). The input and output dimensions of each layer are also given. The light gray areas contain two symmetric procedures.

As can be seen in Fig. 6, two light gray areas include two symmetric procedures, respectively. In the first procedure, the convolutional layers and the max pooling layers are used together to extract the relevant information and to discard the corrupted parts of the CNN input feature vector, resulting in a compression of the vector length. The second procedure is designed to recover the detail information via the combination of upsamping layers and convolutional layers. Meanwhile, the vector length is increased back to the original dimension by using two times the upsampling layer. In the last convolutional layer, a linear activation function is used and the final output has exactly the same dimension L as the input of the CNN. Furthermore, two skip connections are utilized to add up the corresponding layer outputs, in order to ease the vanishing gradient problem during the training of this deep CNN [40].

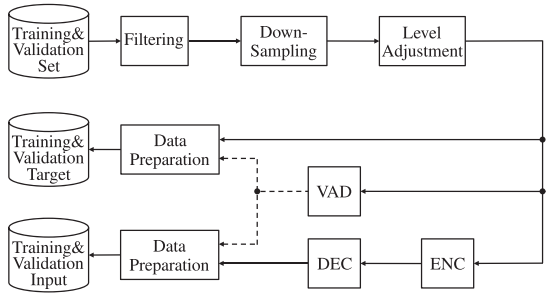


Fig. 7. **Training** and **validation** preprocessing.

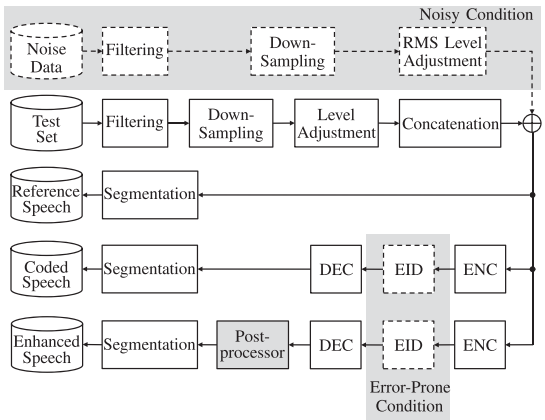


Fig. 8. **Test** processing for various codecs and postprocessors in clean, error-prone transmission, and noisy conditions.

IV. Experimental Setup and Metrics

A. Speech Database

Speech data used in this work is from the NTT wideband speech database [48], containing 21 different languages, and 4 female and 4 male speakers for each language. Each of the speakers is represented by 12 speech utterances of about 8 seconds duration. American English and German are used for test, and for each language, the test set contains 30 speech utterances, in which 3 female speakers and the 3 male speakers are represented by 5 different speech utterances, respectively. For the training set, all speech utterances from 3 female speakers and 3 male speakers *in all other 19 languages* are chosen, while 9 speech utterances from each of the remaining speakers (female speaker `f4` and male speaker `m4` per language) in the same 19 languages are used as validation set. Thereby we provide (partly[3]) language-independent but completely speaker-independent results throughout.

B. Preprocessing for Training and Validation

The training and validation data pairs (i.e., input and target) are obtained following the training and validation preprocessing illustrated in Fig. 7, and the test experiments follow the test processing in Fig. 8. Our training and validation preprocessing and test processing are based upon the original quality assessment plans [49]–[52] for the codecs evaluated in this work and the respective processing functions employed in Figs. 7 and 8 are from the ITU-T software tool library G.191 [53].

The speech utterances are firstly processed by different filters (i.e., FLAT for narrowband codecs[4] and P.341 for wideband codecs). Then, for narrowband codecs the speech signal is decimated from 16 kHz to 8 kHz using the high quality finite impulse response (FIR) low-pass filter HQ2 from [53], while for wideband codecs this downsampling function is bypassed. Then, the active speech level is adjusted to -26 dBov [54].

After this, to obtain the frame indices for the training and validation a very simple frame-based voice activity detection (VAD) is executed as

$$\mathrm{VAD}(\ell) = \begin{cases} 1, \text{ if } \frac{\frac{1}{|\mathcal{N}_\ell|}\sum_{n\in\mathcal{N}_\ell}\tilde{s}^2(n)}{\frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}}\tilde{s}^2(n)} > \theta_{\mathrm{VAD}} \\ 0, \text{ else,} \end{cases} \tag{11}$$

where θ_{VAD} is the VAD threshold, $\mathcal{N}_\ell$ and $\mathcal{N}$ are the sets of sample indices belonging to frame ℓ and the whole speech file, respectively. The frames marked with $\mathrm{VAD}(\ell)=1$ are regarded as active speech frames and the corresponding frame indices are denoted as a set $\mathcal{L}_{\mathrm{VAD}}=\{\ell\,|\,\mathrm{VAD}(\ell)=1\}$. These active speech frames are further used for training and validation, while the other frames are regarded as speech pause and not used in this stage. Then, the target and input for training and validation are obtained as follows:

The *target data* is obtained after the data preparation, in which the windowing w.r.t. the selected time domain or cepstral domain approach is applied to the active speech frames $\ell\in\mathcal{L}_{\mathrm{VAD}}$.

For the *input data of training and validation*, the level-adjusted speech is subject to coding. We examine in total four different speech codecs: two narrowband codecs, which are G.711 [1] and the adaptive differential pulse-code modulation (ADPCM) Recommendation G.726 used for digital enhanced cordless telephony (DECT) at 32 kbps [55], two wideband codecs, which are the wideband ADPCM G.722 used for wideband DECT at 64 kbps [56], and AMR-WB at 12.65 kbps [12] in fixed-point implementation [57] without DTX. The function "ENC" comprises a delay compensation function in case of wideband codecs (cf. assessment plan [52]), a bit conversion function from 16 bits to 14 bits (only for wideband codecs) and the speech encoder from any of the above four codecs. Then, the corresponding function "DEC" is conducted, which comprises the speech decoder, a bit conversion function from 16 bits to 14 bits (only for wideband codecs), and a delay compensation function (only for wideband codecs). Finally, the coded frames with $\ell\in\mathcal{L}_{\mathrm{VAD}}$ form the input data to the data preparation function, which again performs windowing and potential transformation to the cepstral domain.

C. Processing for Training and Validation

In the training processing, we *always* train codec-individual CNN models which are then used later on in test. The prepared input data in the respective domain according to Fig. 7 is at first normalized towards zero mean and unit variance, then this normalized input data and target data is fed into the CNN to train the weights in each convolutional layer. This is achieved

[3]It should be mentioned that British English is one of the 19 training and validation languages, while American English is used in the test. The subjective listening test, however, will be conducted with German samples only, thus being completely language-independent.

[4]Note that for bandwidth consistency reasons, we decided to use the FLAT filter also for G.726 transmission, although typically here an MSIN filter response is used [51], [52].

by minimizing the cost function, which is the mean squared error (MSE) between the outputs of the CNN and the target data. Instead of using the traditional stochastic gradient descent (SGD) algorithm for the trainable weights updating, Adam [58] is used as the learning method to obtain a faster training convergence [40]. In this work, the weights update is performed in each minibatch consisting of 16 frames, being a good trade-off between training speed and performance. At the beginning of each epoch, the training data is shuffled so that the 16 frames of each minibatch are randomly selected from the training data.

In order to train the CNN in an efficient way and to avoid overfitting, the strategies for the learning rate and the stop criteria are the following: The initial learning rate is 5×10^{-4} and it is halved once the MSE on the validation set does not decrease for two epochs. The training stop criterion is checked after each epoch, i.e., after all minibatches have been used, and the training stops if either the MSE on the validation set does not decrease for 16 epochs, or if the number of epochs approaches 100. Finally, the weights are saved as the result of that epoch, after which the lowest MSE on the validation set has been achieved.

D. Processing for Test

In Fig. 8, the test processing functions of filtering, downsampling, level adjustment, "ENC" and "DEC" are identical to those in Fig. 7. Since the proposed postprocessing approaches are evaluated in four conditions, i.e., clean, noisy, tandeming, and error-prone transmission conditions, the test processing is also described for these four conditions. We always select the CNN model that refers to the last employed speech decoder. In most practical applications the last decoder can be assumed to be known, even if in many cases tandem conditions are observed with G.711 being such "last employed decoder". Please note that for the sake of conciseness, we did not include in our simulation the condition when the last decoder is unknown; this could be practically solved by a multi-codec-trained CNN model.

For the *clean condition*, the level-adjusted speech utterances are concatenated to a long speech signal, in which the utterances from female and male speakers are alternately concatenated. After this, the reference speech, coded speech and enhanced speech are obtained as follows:

The reference speech is obtained after segmentation, which cuts the concatenated speech signal back to the original signal portions/durations. Note that this reference speech is also used for the other three conditions.

To obtain coded speech, the function "ENC" and "DEC" are conducted and then the coded speech results after segmentation.

To obtain enhanced speech, the functions "ENC" and "DEC" are conducted and then any of the postprocessors afterwards. Finally the enhanced speech files results after segmentation.

In the *noisy conditions*, three types of noise from the ETSI background noise database [59] are applied in the evaluation part, which are cafeteria noise, car noise at the velocity of 100 km/h, and outside traffic road noise. Similar to the processing of speech utterances in Fig. 8, the noise data is filtered and downsampled or bypassed depending on the codec bandwidth. Then the root mean square (RMS) level of noise is adjusted based on the desired SNR in dB [54]. After this, the adjusted noise is added to the concatenated speech for further processing. Finally, the coded and enhanced speech in the noisy condition are obtained with the same functions as in the clean condition.

In *error-prone transmission conditions*, e.g., mobile and wireless systems, frame losses are inserted to the bitstream after the encoder by using error insertion device (EID) [53], which is placed between the "ENC" and "DEC" in Fig. 8. The coded and enhanced speech in the error-prone transmission conditions are obtained with the other functions being the same as in the clean condition. Two kinds of frame losses are taken into consideration: random frame erasure, which is based on a Gilbert model and burst frame erasure, in which the occurrence of the bursts is modeled by the Bellcore model [53], [60]. Both kinds of frame erasures are characterized by the frame erasure ratio (FER), which is the ratio of the number of distorted frames vs. the number of all transmitted frames.

In *tandeming conditions* we employ a receiver-sided postprocessor for G.711 A-law (narrowband) or the AMR-WB, with various previously mentioned codecs as former codecs, but also the narrowband AMR codec at 12.2 kbps [11], wideband codecs G.711.1 with mode R3 at 96 kbps [3], and EVS-WB at 13.2 kbps [13]. The "EID" block in Fig. 8 is simply replaced by "DEC" and the subsequent "ENC", resulting in a serial connection of two codecs.

E. Metrics of Speech Quality

To instrumentally evaluate the enhanced speech $\hat{s}(n)$, the mean logarithmic spectral distance (LSD) averaged over frames is employed [61]. The LSD is calculated as

$$\mathrm{LSD}(\ell) = \sqrt{\frac{1}{k_{\mathrm{high}} - k_{\mathrm{low}}} \sum_{k=k_{\mathrm{low}}}^{k_{\mathrm{high}}} \left[10 \log_{10}\left(\frac{\left|\tilde{S}(\ell,k)\right|^2}{\left|\hat{S}(\ell,k)\right|^2}\right)\right]^2}, \quad (12)$$

where $\tilde{S}(\ell,k)$ and $\hat{S}(\ell,k)$ and the k-th FFT coefficient of the uncoded and the processed (can be either coded or postprocessed) speech signal in frame ℓ, respectively, and k_{high} and k_{low} are the indices of the upper and lower frequency bin bounds taken into account. The frames used for the mean LSD are from the active speech frame set $\mathcal{L}_{\mathrm{VAD}}$ (from equation (11)) and each frame is formed by employing a 32 ms periodic Hann window with 50% overlap.

To measure the speech distortion, the segmental speech-to-speech-distortion ($\mathrm{SSDR}_{\mathrm{seg}}$) [62] is calculated as

$$\mathrm{SSDR}_{\mathrm{seg}} = \frac{1}{|\mathcal{L}_{\mathrm{VAD}}|} \sum_{\ell \in \mathcal{L}_{\mathrm{VAD}}} \mathrm{SSDR}(\ell), \quad (13)$$

where $\mathrm{SSDR}(\ell)$ is limited from $R_{\min} = -10$ dB to $R_{\max} = 40$ dB by $\mathrm{SSDR}(\ell) = \max\{\min\{\mathrm{SSDR}'(\ell), R_{\max}\}, R_{\min}\}$. The term $\mathrm{SSDR}'(\ell)$ is actually calculated as

$$\mathrm{SSDR}'(\ell) = 10 \log_{10}\left[\frac{\sum_{n\in\mathcal{N}_\ell} \tilde{s}^2(n)}{\sum_{n\in\mathcal{N}_\ell} (\hat{s}(n)-\tilde{s}(n))^2}\right], \quad (14)$$

where $\mathcal{N}_\ell$ is the set of sample indices n belonging to frame ℓ, $\tilde{s}(n)$ and $\hat{s}(n)$ are the uncoded and time-aligned processed (can be either coded or postprocessed) speech signal, respectively. Each frame is also 32 ms with 50% overlap. Note that at some point we will also report on a *global* SSDR measure, which is simply obtained by (14) with setting $\mathcal{N}_\ell = \mathcal{N}$, meaning that all samples in each file contribute to each of the sums in (14). We will call this measure simply SSDR.

For instrumental assessment of speech quality, perceptual evaluation of speech quality (PESQ) [63], [64] for the narrowband speech and WB-PESQ [65] for the wideband speech are

TABLE II
MEAN LOGARITHMIC SPECTRAL DISTANCE (LSD) [DB] ON THE VALIDATION SET. THE BEST SETTING IS WRITTEN IN **BOLD FACE**

	Number of feature maps F							
	20		22		24		26	
	Leaky ReLU	SELU	Leaky ReLU	SELU	Leaky ReLU	SELU	Leaky ReLU	SELU
N = 2	10.77	10.98	10.79	10.76	10.57	10.44	10.74	10.53
N = 4	8.50	8.65	8.54	8.44	8.37	8.65	8.53	8.52
N = 6	8.30	8.44	**8.29**	8.61	8.30	8.42	8.39	8.30
N = 8	8.38	8.44	8.46	8.56	8.45	8.50	8.41	8.47
N = 10	8.46	8.37	8.49	8.44	8.41	8.38	8.41	8.50

used. The output of the two metrics is the mean opinion score (MOS) listening quality objective (LQO), which is denoted as MOS-LQO. A mean value over all test speech utterances for each respective language is reported in the evaluation. In addition to (WB-)PESQ, we also perform perceptual objective listening quality prediction (POLQA) [66]. This is done in only a few conditions, checking, whether both measures lead to similar conclusions.

In addition, for the most promising approaches, we conduct an semi-formal comparison category rating (CCR) subjective listening test according to the ITU-T Recommendation P.800 [67]. In a CCR test, a pair of two speech samples is presented to the listeners, and the quality judgment of the second sample compared to that of the first is made and rated on the comparison MOS (CMOS) scale ranging from -3 (much worse) to $+3$ (much better).

V. Experimental Evaluation and Discussion

In Section V-A, a preliminary experiment is implemented to investigate the CNN topology on the validation set. Then, the optimal setting will be used for the subsequent experiments.

A. Preliminary Experiment on CNN Parameters

In a preliminary experiment the optimal CNN topology settings with the framework structure III of the cepstral domain approach for G.711 postprocessing are selected. The number of feature maps F, the length of the CNN kernels N, and the activation function (the last layer is always linear) are examined. We investigate both leaky rectified linear unit (ReLU) [68] and scaled exponential linear unit (SELU) [69]. Since narrowband speech is used in this preliminary experiment, a frequency region from 50 Hz to 3.4 KHz is taken into account, resulting in $k_{\text{high}} = \lfloor \frac{K}{8000} \cdot 3400 \text{ Hz} \rfloor = 217$ and $k_{\text{low}} = \lfloor \frac{K}{8000} \cdot 50 \text{ Hz} \rfloor = 3$ in equation (12) with the 512-point FFT.

The results are shown in Table II, in which we can see that the performance of the CNN in our proposed approach is mainly depending on the kernel length N, and only weakly on the choice of the activation function and the number of feature maps F. Note that only a small fraction of the actually used (N, F) search space is shown in Table II: The dependence on F regarding to the optimum is rather flat, however, for much smaller values of F the performance deteriorates significantly.

As a result, the CNN topology with the minimum mean LSD value of 8.29 dB recommends the choices $F_{\text{opt}} = 22$, $N_{\text{opt}} = 6$, and the leaky ReLU activation function. It is interesting to know that the legacy G.711 has a mean LSD being 16.15 dB which is almost halved by applying this optimal topology. Note that F_{opt} and N_{opt} selected from the above preliminary experiment are specific to the framework structure III with the length $L = 32$ of the CNN input vector. In order to obtain also reasonable parameter settings for the other framework structures, we note that the length L changes for the various postprocessing approaches with $L = |\mathcal{M}_{\text{env}}| = 6.25\% \cdot K$ in the cepstral domain approaches, and $L = 80$ for narrowband codecs and $L = 160$ for wideband codecs in the time domain approach. Note that for simplicity of presentation, whenever L changes with a certain framework structure (time domain, cepstral domain I–VI), the value of F_{opt} and N_{opt} are simply increased or decreased proportionally at the same time.

TABLE III
LSD [DB] VALUES ON THE VALIDATION SET AND THE NUMBER OF TRAINABLE PARAMETERS (# OF PARAM.) FOR THE OPTIMAL CNN WITH $(F_{\text{opt}}, N_{\text{opt}})$ AND FOUR DIFFERENT FULLY-CONNECTED NEURAL NETWORKS WITH OR WITHOUT DROPOUT (DROPOUT RATE r) IN TIME DOMAIN AND CEPSTRAL DOMAIN. THE TOPOLOGIES YIELDING THE LOWEST LSD VALUES ARE WRITTEN IN **BOLD FACE**

Time Domain				Cepstral Domain			
Topology	r	# of Param.	LSD [dB]	Topology	r	# of Param.	LSD [dB]
CNN $(F_{\text{opt}}, N_{\text{opt}})$	-	**0.82M**	**11.03**	CNN $(F_{\text{opt}}, N_{\text{opt}})$	-	**52.82K**	**8.29**
1024-1024	0	1.21M	12.12	256-256	0	82.46K	9.59
	0.1		14.71		0.1		9.54
512-512-1024	0	0.91M	12.33	128-128-256	0	61.98K	9.89
	0.1		28.90		0.1		9.63
512-512-512 -512	0	0.87M	13.08	128-128-128 -128	0	57.89K	9.70
	0.1		30.68		0.1		11.14
512-512-256 -512-512	0	0.87M	12.03	128-128-64 -128-128	0	57.95K	9.24
	0.1		55.03		0.1		15.53

Now, as we have fixed the number of trainable parameters, we briefly want to check whether a straight-forward fully-connected neural network (FCNN) performs equally well. As shown in Table III, we simulated four different FCNN topologies without dropout or a dropout rate $r = 0.1$ for both the time domain approach and the cepstral domain approach of structure III, while keeping the same number of input nodes ($L = 80$ for the time domain approach and $L = 32$ for the cepstral domain approach). The number of trainable parameters is about the same as (or a bit higher than) the optimal CNN topology with $(F_{\text{opt}}, N_{\text{opt}})$. It can be seen that the optimal CNN topology achieves the best LSD performance compared to all listed FCNN structures for both the time domain approach and the cepstral domain approach. Accordingly, in the following we stick to the CNN topology as it seems to be an advantageous choice.

B. Major Instrumental Experiments

In this subsection, the experiments of the proposed postprocessing approaches for the various codecs in different conditions are implemented and evaluated instrumentally following the test processing in Fig. 8.

1) Clean Condition: A comprehensive evaluation of all the proposed postprocessors is conducted for four different codecs in both American English and German language, in which legacy codecs and the postfilter for G.711 serve as baselines. PESQ results are shown in Table IV with ΔMOS-LQO being the MOS-LQO difference between the postfilter or the postpro-

TABLE IV
MOS-LQO (**PESQ** AND **WB-PESQ**) FOR LEGACY CODECS AND CODECS WITH VARIOUS POSTPROCESSORS. THE **TOP TWO** RESULTS IN EACH COLUMN ARE WRITTEN IN **BOLD FACE**

			American English					German				
			G.711 A-law no Constr.	G.711 A-law Constr.	G.726	G.722	AMR-WB	G.711 A-law no Constr.	G.711 A-law Constr.	G.726	G.722	AMR-WB
Legacy Codec		MOS-LQO	4.21		3.96	3.72	3.60	4.15		4.01	3.61	3.53
Postfilter [2]		MOS-LQO	4.32		-	-	-	4.25		-	-	-
		ΔMOS-LQO	0.11					0.10				
Time Domain		MOS-LQO	4.32	4.32	4.21	4.32	3.61	4.30	4.30	4.26	4.29	3.62
		ΔMOS-LQO	0.11	0.11	0.25	0.60	0.01	0.15	0.15	0.25	0.68	0.09
Cepstral Domain	I	MOS-LQO	4.24	4.27	3.99	4.13	3.45	4.13	4.18	4.01	4.07	3.29
		ΔMOS-LQO	0.03	0.06	0.03	0.41	-0.15	-0.02	0.03	0	0.46	-0.24
	II	MOS-LQO	4.40	4.30	4.15	**4.47**	3.78	4.39	4.24	4.26	4.46	3.73
		ΔMOS-LQO	0.19	0.09	0.19	**0.75**	0.18	0.24	0.09	0.25	0.85	0.20
	III	MOS-LQO	**4.43**	4.33	4.20	**4.47**	**3.79**	**4.42**	4.26	4.29	**4.48**	**3.74**
		ΔMOS-LQO	**0.22**	0.12	0.24	**0.75**	**0.19**	**0.27**	0.11	0.28	**0.87**	**0.21**
	IV	MOS-LQO	4.27	4.27	4.01	4.17	3.52	4.17	4.19	4.04	4.12	3.41
		ΔMOS-LQO	0.06	0.06	0.05	0.45	-0.08	0.02	0.04	0.03	0.51	-0.12
	V	MOS-LQO	4.42	4.31	**4.21**	4.45	3.74	4.41	4.26	**4.30**	4.44	3.67
		ΔMOS-LQO	0.21	0.10	**0.25**	0.73	0.14	0.26	0.11	**0.29**	0.83	0.14
	VI	MOS-LQO	**4.44**	4.31	**4.25**	**4.50**	**3.85**	**4.42**	4.23	**4.32**	**4.47**	**3.79**
		ΔMOS-LQO	**0.23**	0.10	**0.29**	**0.78**	**0.25**	**0.27**	0.08	**0.31**	**0.86**	**0.26**

TABLE V
MOS-LQO (**POLQA**) FOR LEGACY CODECS AND CODECS WITH ITU-T POSTFILTER [2] AND THE STRUCTURE III CEPSTRAL DOMAIN POSTPROCESSOR. THE BEST RESULTS IN EACH COLUMN ARE WRITTEN IN **BOLD FACE**. COMPARE TO THE RESPECTIVE (WB-)PESQ RESULTS IN TABLE IV

		American English					German				
		G.711 A-law no Constr.	G.711 A-law Constr.	G.726	G.722	AMR-WB	G.711 A-law no Constr.	G.711 A-law Constr.	G.726	G.722	AMR-WB
Legacy Codec	MOS-LQO	4.29		4.03	3.78	3.64	4.14		4.03	3.71	3.65
Postfilter [2]	MOS-LQO	4.46		-	-	-	4.33		-	-	-
	ΔMOS-LQO	0.17					0.19				
Structure III	MOS-LQO	**4.49**	**4.40**	**4.31**	**4.73**	**3.97**	**4.45**	**4.27**	**4.38**	**4.56**	**3.93**
Cepstral Domain	ΔMOS-LQO	**0.20**	**0.11**	**0.28**	**0.95**	**0.33**	**0.31**	**0.13**	**0.35**	**0.85**	**0.28**

cessor and the respective legacy codec. We find that most of our proposed postprocessors perform better than the respective legacy codecs. For G.711 our proposed postprocessors in most cases show better performance when no quantization constraint is performed. Comparing the various proposed postprocessors with no quantization constraint, the time domain postprocessor and the cepstral domain postprocessors with structures II, III, V, and VI (the ones with delay, see Table I) show *better performance than all legacy codecs* and they *all perform better than or equal to the G.711 postfilter* [2] for both languages (only the time domain postprocessor has the same MOS-LQO as the postfilter for American English). The cepstral domain postprocessor with structure VI performs best for both languages and for all codecs, exceeding the legacy codecs on average over both languages by 0.25 MOS points for G.711, 0.3 MOS points for G.726, and 0.26 MOS points for AMR-WB. Note that structure VI exceeds the G.722 legacy codec by an impressive 0.82 MOS points, where roughly 0.3 MOS points can be dedicated to the rather simple suppression of frequencies beyond 7 kHz, and the major rest can be dedicated indeed to the improvement of the early cepstral coefficients.

For a limited set of conditions in Table IV, we provide also POLQA [66] results in Table V. Note that very similar improvements of our postprocessor (structure III) w.r.t. all legacy codecs in both languages can be seen, with the AMR-WB postprocessor performing even better in POLQA than in WB-PESQ. However, since simulation of PESQ was much easier to perform due to the availability of a batch mode to us, the remainder of our work uses PESQ and WB-PESQ.

In order to obtain a better understanding of how the coded speech signal is enhanced by the cepstral domain approach, spectral and cepstral analysis examples of the enhanced speech, along with the coded and reference speech, are presented for interested readers in the Appendix.

Since the algorithmic delay might be critical in practical applications, we see that the zero-latency time domain postprocessors can improve the speech quality for all listed codecs in both languages. For cepstral domain postprocessors, the zero-latency structures I and IV still can consistently improve speech quality of G.726 and particularly of G.722. Since G.711 and AMR-WB ask for some delay in the postprocessor, a good compromise for these codecs would be the structure III, providing second

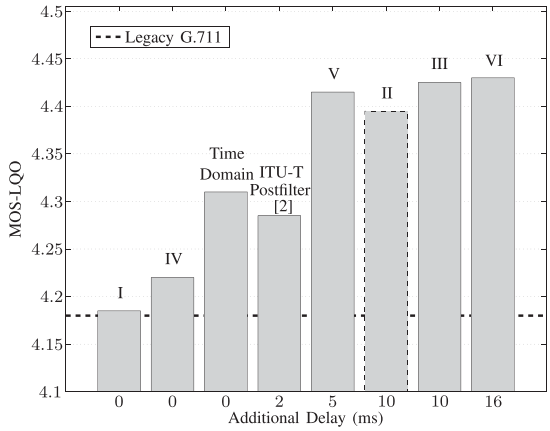


Fig. 9. The MOS-LQO (**PESQ**) of various postprocessors for G.711 with different amounts of additional delay. Note that structure II has a different model topology than the other cepstral domain structures I, III–VI; see Table XII.

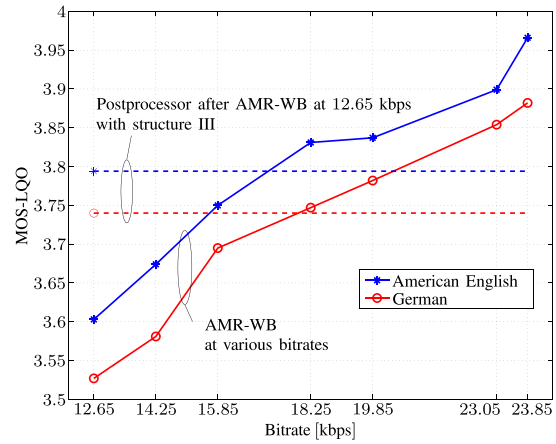


Fig. 10. MOS-LQO (**WB-PESQ**) points of the test speech utterances for the legacy AMR-WB at various bitrates (solid curves from 12.65 to 23.85 kbps) as well as for our postprocessor with structure III at 12.65 kbps (dashed lines). Results are shown for American English (∗) and German (○).

ranked speech quality in both languages. At the cost of only 10 ms algorithmic delay, structure III exceeds the legacy codecs on average over both languages by 0.25 MOS points for G.711, 0.25 MOS points for G.726, 0.81 MOS points for G.722, and 0.2 MOS points for AMR-WB.

To further illustrate the influence of the additional delay on the performance improvement, we compare in Fig. 9 the MOS-LQO of the postfilter and the postprocessors both in time domain and cepstral domain for G.711, sorted by the additional delay. The MOS-LQO is an average of American English and German. For the proposed postprocessors in the cepstral domain, it becomes obvious that the performance improvement grows with the increase of the additional delay, as the model topology is exactly the same (except for structure II,[5] see Table XII). With longer additional delay for the postfilter [2], it may also achieve some further performance gains. However, our proposed zero-latency postprocessor in the time domain already shows superior performance compared to the ITU-T postfilter with 2 ms additional delay.

Comparing the bold face (i.e., top-two) results in Table IV, we see that there is hardly a language dependency in the rank order of the best approaches.

To intuitively show the potential of the postprocessor with structure III we performed a comparison to different modes (i.e., bitrates) for the AMR-WB codec in Fig. 10. One can easily see that the MOS-LQO of the postprocessor after the AMR-WB codec at 12.65 kbps for both American English and German exceeds the legacy AMR-WB at 15.85 kbps and it even approaches a comparable quality for German at 18.25 kbps. Therefore, the postprocessor with structure III shows its ability to significantly improve the speech quality during transmission with a relative low bitrate towards a much higher bitrate transmission.

In order to see the waveform distortion of speech after the postprocessing, SSDR_{seg} measure (13) for G.711 A-law in American English and German is shown in Table VI. It is straightforward that the legacy G.711 already achieves a relatively high SSDR_{seg}, with 37.12 dB for American English and 37.11 dB for German, since it is a high bitrate waveform coding. For the time domain postprocessor, it achieves even higher SSDR_{seg} which is the best performance among all the proposed postprocessors, since it focuses on the waveform domain. All proposed postprocessors with quantization constraint show equal or better SSDR_{seg} than without it, but it brings no positive effect to speech quality for the proposed postprocessors in terms of MOS-LQO (see Table IV). For the postprocessor with structure VI, which achieves the best speech quality (see Table IV), a mean SSDR_{seg} of only 24.04 dB over both languages is measured. Comparing SSDR_{seg} and MOS-LQO, we once again see that waveform similarity and speech quality are not necessarily positively correlated, in this case also questioning the quantization constraint.

TABLE VI
THE SSDR_{SEG} [DB] VALUES FOR THE G.711 LEGACY CODEC AND G.711 CODEC WITH VARIOUS POSTPROCESSORS. THE BEST APPROACH IS WRITTEN IN **BOLD FACE**

		American English		German	
		no Constr.	Constr.	no Constr.	Constr.
Legacy Codec		37.12		37.11	
Postfilter [2]		17.27		15.65	
Time Domain		**38.07**	**38.08**	**38.15**	**38.15**
Cepstral Domain	I	29.37	29.98	29.42	29.99
	II	21.55	29.93	21.80	29.94
	III	25.85	29.96	26.23	29.97
	IV	29.36	29.98	29.42	29.98
	V	26.67	29.97	26.91	29.98
	VI	23.75	29.95	24.33	29.95

Evaluating the global SSDR measure in Table VII, it turns out that the rank order of approaches is very similar to the SSDR_{seg} in Table VI: The proposed time domain approach is the best, followed by the G.711 legacy codec, the cepstral domain approaches, and finally the ITU-T postfilter [2]. Interestingly, the advantage of using the constraint is higher with the SSDR measure, which might be due to some very slight residual noise for the cepstral domain approaches in speech pauses; an effect that has been disregarded in SSDR_{seg} through the inherent voice activity detection in (13), and which will motivate some small extra processing in Section V-C.

[5]This is because structure II has a different topology in terms of the number of input nodes L, feature maps F, and kernel size N (see Table XII).

TABLE VII
THE **SSDR [DB]** VALUES FOR THE G.711 LEGACY CODEC AND G.711 CODEC WITH VARIOUS POSTPROCESSORS. THE BEST APPROACH IS WRITTEN IN **BOLD FACE**

		American English		German	
		no Constr.	Constr.	no Constr.	Constr.
Legacy Codec		37.30		37.35	
Postfilter [2]		17.79		15.70	
Time Domain		**38.33**	**38.34**	**38.49**	**38.50**
Cepstral Domain	I	29.98	34.15	30.05	34.17
	II	19.94	32.36	20.54	32.35
	III	24.58	33.04	25.14	33.03
	IV	29.93	34.11	30.11	34.15
	V	25.39	32.97	25.83	32.93
	VI	21.80	32.66	23.26	32.71

2) Tandeming Conditions: In order to evaluate the performance of the proposed postprocessors in tandeming conditions, G.711 A-law and AMR-WB are selected as the last codec for narrowband and wideband, respectively, while several other codecs form some common tandeming conditions. The CNN model matches the last codec since only this codec is known at the receiving point. It is worth noting that all further experiments in this subsection are only conducted in American English. The PESQ results are shown in Table VIII and we can see the performance of our time domain postprocessor and the postprocessor in the cepstral domain with structures III and VI. While in narrowband tandem conditions structure III achieves a MOS-LQO improvement in the range 0.11...0.20 points (in all cases the postprocessor has been just trained for the receiving-sided A-law G.711), the structure III in wideband tandeming conditions improves by 0.20...0.37 PESQ MOS points (the postprocessor has been only trained for the receiving-sided AMR-WB). Note that the G711.1 A-law + AMR-WB tandeming and the G.722 + AMR-WB tandeming, followed by structure III both achieve around 3.7 PESQ MOS points, which is even more than only AMR-WB with 3.6 points (see Table IV). With the best postprocessor of structure VI from Table IV, even slightly better speech quality is achieved in all cases for the price of a large algorithmic delay. *All of the postprocessors in Table VIII exceed the shown legacy codecs under tandeming, even if the legacy codec (G.711 A-law) is followed by the postfilter from [2].*

3) Error-Prone Transmission Conditions: For the evaluation of the proposed postprocessors in error-prone transmission, random and burst frame losses are inserted to the bitstream of G.711 and AMR-WB with the FER being 3% and 6%and the PESQ results are shown in Table IX. It is worth noting that the error concealment measures are applied in all conditions for both codecs: the packet loss concealment for G.711 from Appendix I [70] and the error concealment of erroneous or lost frames for AMR-WB from 3GPP TS 26.191 [71]. Note that AMR-WB in this condition requires DTX to be switched on. The time domain postprocessor has better or equal performance compared to the postfilter [2] for G.711 for both random and burst frame losses, and is very slightly better in the case of AMR-WB. The cepstral domain postprocessors with structures III and VI both perform even better in all cases and structure III with less delay improves the legacy codecs by 0.06...0.16 PESQ MOS points in narrowband frame loss and 0.14...0.19 PESQ MOS points in wideband frame loss. *Accordingly, all of the postprocessors in Table IX can be advantageously employed after the legacy codecs in frame loss conditions.*

4) Noisy Speech Conditions: In order to evaluate the performance of the proposed postprocessing approaches for noisy speech, different types of background noise are added to the speech signals at an SNR of 15 dB or 20 dB, followed by G.711 and AMR-WB. The PESQ results are shown in Table X, while the mean of the noisy conditions and also the clean conditions for both codecs are listed. For the G.711-based narrowband experiments with noisy speech, both the postfilter [2] and the proposed postprocessors hardly have an influence on the coded speech, with MOS-LQO differences being less than 0.04, and two insignificant degradations of only 0.01 MOS points being observed. On average, the postfilter and the proposed postprocessors have a MOS-LQO improvement in the range 0...0.02 points. For AMR-WB in noisy conditions, the cepstral domain postprocessors can improve or maintain the speech quality for most of the cases, with two exceptions: cafeteria noise (0.01 MOS points decrease) and road noise (0.03 MOS points decrease) both at 15 dB SNR. For car noise at both 15 and 20 dB obviously a speech quality improvement has been observed: 0.13 and 0.16 MOS points for structure III, 0.17 and 0.24 MOS points for structure VI. The means over the noisy conditions show a MOS-LQO improvement of 0.05 points for structure III and 0.10 points for structure VI. In summary and on average, *both the G.711 postfilter and our proposed postprocessors do neither significantly improve nor distort noisy speech quality at the receiver.*

Finally, in order to increase the robustness of the approach, we also trained the structure III model jointly with clean and noisy speech data. Four noise types[6] from the QUT-NOISE database [72] (noise types are different to the test data) are used to generate the 20 dB noisy training data, with the amount of the noisy data being one quarter of the clean data. As can be seen in Table X, the model trained with noisy data (in the grey-shaded rows) achieves best performance on average and over the noisy conditions for both G.711 and AMR-WB. A test on clean data expectedly shows some reduced performance improvement. In summary and on average, *the proposed postprocessor trained with additional noisy data can provide even some improvements in noisy conditions.*

C. Subjective Experiment

In our CCR subjective listening test, 2 female and 12 male listeners participated, who are native German speakers stating to have no hearing impairment. An amount of 16 utterances from 4 speakers (2 female and 2 male) of the NTT speech database in German are subject to four test conditions following the processing plan in clean condition of Fig. 8: The first is the *direct* condition, resulting in the reference speech. The second is the *legacy G.711* condition, providing speech transcoded by the G.711 codec. The third is the *postfilter* condition, where G.711-transcoded speech has been enhanced by the ITU-T postfilter [2]. The fourth is the *proposed postprocessor* condition, where G.711-transcoded speech has been enhanced by our proposed postprocessor of structure III in the cepstral domain. Finally, all speech signals are converted to 48 kHz sampling rate.

[6]The four noise types are: HOME-KITCHEN, HOME-LIVINGB, REVERB-POOL, and REVERB-CARPARK.

TABLE VIII
MOS-LQO (**PESQ** AND **WB-PESQ**) FOR LEGACY CODECS AND CODECS WITH DIFFERENT POSTPROCESSORS IN **TANDEMING CONDITIONS**. THE RESULTS OF THE BEST APPROACH IS WRITTEN IN **BOLD FACE**

		Narrowband Tandeming			Wideband Tandeming		
		μ-law + A-law	G.726 + A-law	AMR + A-law	G.711.1 (A-law) + AMR-WB	G.722 + AMR-WB	EVS-WB + AMR-WB
Legacy Codec	MOS-LQO	4.18	3.96	4.01	3.37	3.34	3.28
Postfilter [2]	MOS-LQO	4.20	4.01	4.07	-	-	-
	ΔMOS-LQO	0.02	0.05	0.06			
Time Domain	MOS-LQO	4.28	4.03	4.09	3.39	3.49	3.29
	ΔMOS-LQO	0.10	0.07	0.08	0.02	0.15	0.01
Structure III	MOS-LQO	4.38	4.13	4.12	3.70	3.71	3.48
Cepstral Domain	ΔMOS-LQO	0.20	0.17	0.11	0.33	0.37	0.20
Structure VI	MOS-LQO	**4.41**	**4.18**	**4.13**	**3.78**	**3.75**	**3.53**
Cepstral Domain	ΔMOS-LQO	**0.23**	**0.22**	**0.12**	**0.41**	**0.41**	**0.25**

TABLE IX
MOS-LQO (**PESQ** AND **WB-PESQ**) FOR G.711 AND AMR-WB LEGACY CODECS AND CODECS WITH DIFFERENT POSTPROCESSORS IN **ERROR-PRONE TRANSMISSION CONDITIONS**. THE RESULTS OF THE BEST APPROACH IS WRITTEN IN **BOLD FACE**

		G.711				AMR-WB			
		Random		Burst		Random		Burst	
		3%	6%	3%	6%	3%	6%	3%	6%
Legacy Codec	MOS-LQO	3.67	3.31	3.60	3.07	2.75	2.30	2.80	2.39
Postfilter [2]	MOS-LQO	3.71	3.34	3.66	3.12	-	-	-	-
	ΔMOS-LQO	0.04	0.03	0.06	0.05				
Time Domain	MOS-LQO	3.71	3.35	3.67	3.12	2.78	2.32	2.83	2.41
	ΔMOS-LQO	0.04	0.04	0.07	0.05	0.03	0.02	0.03	0.02
Structure III	MOS-LQO	3.74	3.37	**3.76**	**3.19**	2.94	2.44	2.99	2.54
Cepstral Domain	ΔMOS-LQO	0.07	0.06	**0.16**	**0.12**	0.19	0.14	0.19	0.15
Structure VI	MOS-LQO	**3.76**	**3.41**	3.73	3.16	**3.03**	**2.51**	**3.09**	**2.62**
Cepstral Domain	ΔMOS-LQO	**0.09**	**0.10**	0.13	0.09	**0.28**	**0.21**	**0.29**	**0.23**

TABLE X
MOS-LQO (**PESQ** AND **WB-PESQ**) FOR G.711 AND AMR-WB LEGACY CODECS AND CODECS WITH DIFFERENT POSTPROCESSORS IN **NOISY SPEECH CONDITIONS**. THE RESULTS OF THE BEST APPROACH IS WRITTEN IN **BOLD FACE** AND THE MODEL TRAINED WITH 20 dB (UNSEEN) NOISY DATA IS GREY-SHADED

		G.711								AMR-WB							
		Cafeteria		Car		Road		Mean	Clean	Cafeteria		Car		Road		Mean	Clean
SNR [dB]		15	20	15	20	15	20			15	20	15	20	15	20		
Legacy Codec	MOS-LQO	2.29	2.67	2.40	2.75	2.06	2.43	2.43	4.21	1.69	2.10	2.12	2.52	1.59	1.99	2.00	3.60
Postfilter [2]	MOS-LQO	**2.31**	2.70	**2.41**	**2.76**	2.07	2.45	2.45	4.32	-	-	-	-	-	-	-	-
	ΔMOS-LQO	**0.02**	0.03	**0.01**	**0.01**	0.01	0.02	0.02	0.11								
Time Domain	MOS-LQO	**2.31**	2.69	**2.41**	**2.76**	2.06	2.45	2.45	4.32	1.73	1.74	1.98	2.39	1.63	2.05	1.92	3.61
	ΔMOS-LQO	**0.02**	0.02	**0.01**	**0.01**	0	0.02	0.02	0.11	0.04	-0.36	-0.14	-0.13	0.04	0.06	-0.08	0.01
Structure III	MOS-LQO	2.29	2.68	2.39	2.75	2.05	2.43	2.43	4.43	1.68	2.13	2.25	2.68	1.56	1.99	2.05	3.79
Cepstral Domain	ΔMOS-LQO	0	0.01	-0.01	0	-0.01	0	0	0.22	-0.01	0.03	0.13	0.16	-0.03	0	0.05	0.19
Structure III	MOS-LQO	**2.31**	**2.73**	2.38	2.70	**2.19**	**2.60**	**2.49**	4.32	**1.75**	**2.19**	2.24	2.63	**1.80**	**2.22**	**2.14**	3.66
Cepstral Domain	ΔMOS-LQO	**0.02**	**0.06**	-0.02	-0.05	**0.13**	**0.17**	**0.06**	0.11	**0.06**	**0.09**	0.12	0.11	**0.21**	**0.23**	**0.14**	0.06
Structure VI	MOS-LQO	2.30	2.69	2.40	2.75	2.06	2.45	2.44	**4.44**	1.71	2.17	**2.29**	**2.76**	1.59	2.05	2.10	**3.85**
Cepstral Domain	ΔMOS-LQO	0.01	0.02	0	0	0	0.02	0.01	**0.23**	0.02	0.07	**0.17**	**0.24**	0	0.06	0.10	**0.25**

TABLE XI
CCR SUBJECTIVE LISTENING TEST RESULTS WITH THE BASELINE POSTFILTER [2], THE PROPOSED POSTPROCESSOR OF STRUCTURE III, THE LEGACY G.711 CODEC AND THE DIRECT CONDITION. THE WINNING CONDITION IS WRITTEN IN **BOLD FACE**

CCR Cases	CMOS	CI_{95}
Legacy G.711 vs. **Direct**	1.76	[1.61; 1.92]
Postfilter [2] vs. **Direct**	0.28	[0.13; 0.43]
Proposed Postprocessor vs. Direct	-0.18	[-0.33; -0.02]
Legacy G.711 vs. **Postfilter [2]**	1.45	[1.27; 1.64]
Legacy G.711 vs. **Proposed Postprocessor**	1.77	[1.60; 1.95]
Postfilter [2] vs. **Proposed Postprocessor**	0.36	[0.23; 0.50]

These four conditions result in six comparison cases in the subjective listening test (cf. Table XI).

In a preliminary informal subjective listening test we observed that an ideally very low 0-th cepstral coefficient turns out to assume slightly higher values after the CNN estimation, resulting in somewhat noisy speech pauses. Therefore, for the subjective listening test, we very slightly manipulate the CNN output as follows[7]

$$\hat{c}_{\text{env}}(\ell,0) \rightarrow \begin{cases} \hat{c}_{\text{env}}(\ell,0), \text{ if } \hat{c}_{\text{env}}(\ell,0) \geqslant C_0 \\ \hat{c}_{\text{env}}(\ell,0) - \gamma_0, \text{ else,} \end{cases} \tag{15}$$

with $C_0 = -1650$ and $\gamma_0 = 1000$.

The participants of the subjective listening test rated the speech using an `AKG K-271 MKII` headphone from a computer with external `RME Fireface 400` sound card. The participants were equally assigned to one of two disjoint sets, where the speech is balanced over the comparison cases and the speakers. Each participant familiarized himself with all the comparison cases and was asked to choose a proper volume on the basis of 12 sample pairs in the familiarization phase. Then, each participant evaluated 72 sample pairs in the main test phase, where 36 sample pairs are presented in both sample orders.

In Table XI, the CMOS and respective 95% confidence interval (CI_{95}) for the six CCR comparison cases are shown. *All results turned out to be significant*. We can see a clear 1.76 CMOS points advantage for the comparison of legacy G.711 vs. direct. For the cases where the direct condition is compared to the postfilter [2] and the proposed postprocessor of structure III in the cepstral domain, 0.28 and -0.18 CMOS points are obtained, respectively. This means that the speech enhanced by the proposed postprocessor is more similar to the uncoded speech (in direct condition), and even slightly but *significantly preferred to uncoded speech*. To the best knowledge of the authors, such a result has never been reported before. For details, however, see [73]. Our only explanation is the very low-energy in speech pause during the direct condition, which, of course, we are not allowed to manipulate. Relative to the legacy G.711 condition, the ITU-T postfilter [2] already shows a significant 1.45 CMOS points advantage, while the proposed postprocessor performs even better, obtaining 1.77 CMOS points above the legacy G.711 condition. When the proposed postprocessor is directly compared to the ITU-T postfilter [2], a better performance of 0.36 CMOS points is obtained. Finally, we conclude that the proposed postprocessor improves the quality of G.711-coded speech more effectively as the ITU-T postfilter [2] does.

[7]Note that this manipulation naturally also degrades the instrumental values as given in Section V-B1. For structure III in Table IV, e.g., we observed deviations in the range $[-0.12 \ldots +0.02]$ over languages and codecs, however, still exceeding all legacy codecs and the postfilter [2] in instrumental metrics.

TABLE XII
COMPUTATIONAL COMPLEXITY IM MIPS FOR THE DOMINANT CONVOLUTIONAL OPERATIONS IN THE CNN OF EACH PROPOSED FRAMEWORK STRUCTURE IN NARROWBAND. THE NUMBER OF FRAMES PER SECOND AND THE PARAMETERS OF THE CNN FOR ALL THE PROPOSED FRAMEWORK STRUCTURES (L, N, AND F) ARE ALSO LISTED

		Frames per second	L	N	F	MIPS
Time Domain		100	80	15	55	3820 (!)
Cepstral Domain	I	100	32	6	22	98.4
	II	200	16	3	11	12.4
	III	100	32	6	22	98.4
	IV	50	32	6	22	49.2
	V	50	32	6	22	49.2
	VI	62.5	32	6	22	61.5

D. Complexity Analysis

The complexity of the time domain approach basically lies in the computations for the CNN. Neglecting the operations in Fig. 6 of max pooling, upsampling and skip connection addition, the complexity-dominant convolutional operations of the CNN amount to about $10.5 \cdot NLF^2 + 2 \cdot NLF$ multiply/accumulates (MACs) per frame of the time domain approach, with L being the frame length (i.e., 10 ms of speech samples) and N, F being the parameters of CNN. For the cepstral domain approaches, the number of MACs in the CNN follows the same expression as the time domain approach, with $L = |\mathcal{M}_{\text{env}}| = 6.25\% \cdot K$. Moreover, some operations are required besides the computations in the CNN: FFT, IFFT, DCT-II, and IDCT-II all have a computational complexity of $\mathrm{O}(K \log K)$ [74], [75].

In order to show the complexity of the proposed CNN-based postprocessors, the million instructions (= MACs) per second (MIPS) for the convolutional operations in the CNN of each proposed framework structure in narrowband are shown in Table XII. Note that the values of L, N, and F are doubled in wideband, resulting in a larger number of MACs per second compared to that in narrowband. We see that the time domain postprocessor requires a lot of computations, while the cepstral domain postprocessors have moderate complexity in terms of MIPS, roughly in the order of magnitude of a modern speech codec. As an outlook to future work, however, it might be attractive to reduce the complexity of the models further by methods such as teacher-student learning.

VI. CONCLUSION

In this work, we propose two different CNN-based postprocessing approaches in the time domain and the cepstral domain, including six different framework structures for the latter, to enhance coded speech in a system-compatible manner. The proposed postprocessors in both domains are evaluated for various narrowband and wideband speech codecs in clean, tandeming, error-prone transmission and noisy conditions, and they are compared to an ITU-T postfilter [2] as postprocessing baseline for G.711. The proposed postprocessor improves

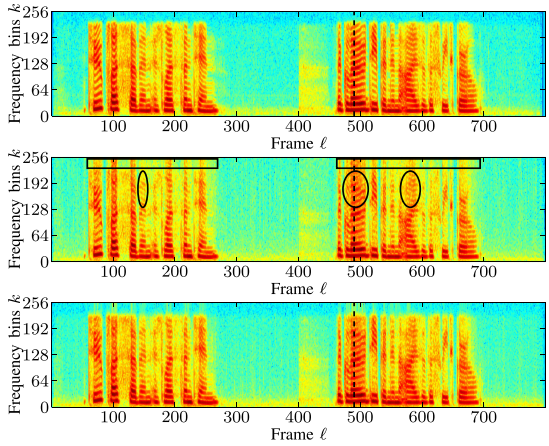


Fig. 11. Narrowband spectrograms of an utterance: reference speech (top), G.726-coded speech (center), and postprocessed speech (bottom). Characteristic time-frequency regions and frame $\ell = 490$ are marked.

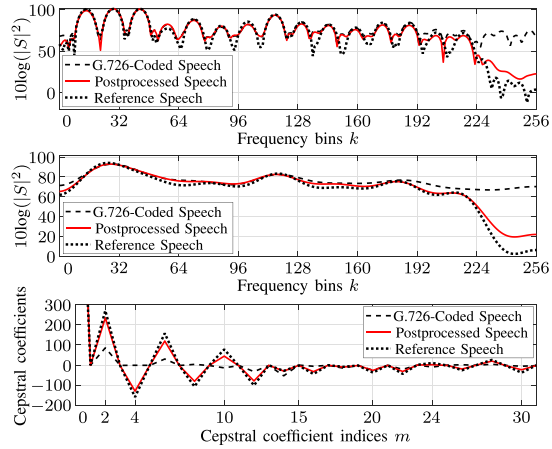


Fig. 12. Amplitude spectrum (top), spectral envelope (center), both on a logarithmic scale, and DCT-II type of cepstral coefficients (bottom) for frame $\ell = 490$ (see Fig. 11) of the narrowband reference speech, G.726-coded speech, and postprocessed speech, respectively.

speech quality in terms of PESQ by up to 0.25 MOS-LQO points for G.711, 0.30 points for G.726, 0.82 points for G.722, and 0.26 points for AMR-WB. In a subjective CCR listening test, the proposed postprocessor on G.711-coded speech exceeds the speech quality of an ITU-T-standardized postfilter by 0.36 CMOS points, and obtains a clear preference of 1.77 CMOS points compared to G.711, even significantly exceeding the quality of uncoded speech. The source code for the cepstral domain approach to enhance G.711-coded speech is available at https://github.com/ifnspaml/Enhancement-Coded-Speech.

Appendix

In this Appendix we will provide some further detailed analysis of our postprocessor in certain conditions.

We take the speech file `am02f065` from the NTT speech database in American English as an example, and plot the spectrograms for preprocessed speech (i.e., reference speech, see Fig. 8), G.726-coded speech, and enhanced speech by the cepstral domain approach with structure III in Fig. 11. The spectral analysis settings are identical to the framework structure III (see Table I). Comparing top and center subplots, the G.726 coding adds signal contents to the high frequencies (marked by rectangles) and distorts/weakens spectral envelope (marked by ovals). For the enhanced speech in the bottom, the high frequency coding noise is effectively eliminated, and the spectral envelope is somewhat being restored and enhanced towards the reference speech spectral envelope.

In order to show that the improvement of the postprocessor is not only based on a trivial postfilter simply suppressing frequencies beyond 3.5 kHz, we did a brief PESQ MOS measurement of coded speech with high frequencies simply removed: A lowpass filter cutting off at 3.5 kHz is applied to the coded speech. The FLAT filter is used here, along with the up- and downsampling, since the FLAT filter works at 16 kHz. It turns out that the PESQ MOS scores did not even change after this trivial postfiltering for both narrowband codecs (G.711 and G.726) for both languages. This maybe surprising result shows that the major speech quality improvement does not at all come from the trivial filtering, but supports our proposed postprocessor which also acts on lower frequencies.

As the proposed cepstral domain approach intends to improve the spectral envelope, we zoom into the frame $\ell = 490$ (dashed line in Fig. 11) to have a clear view of the spectral envelope. In Fig. 12, the logarithmic spectrum $10\log(|S(k)|^2)$ of the selected frame is drawn in the top, and of the spectral envelope in the middle, obtained by keeping the first 32 cepstral coefficients and setting the other cepstral coefficients to zero, i.e., lowpass liftering. As we can see, the spectral envelope of the enhanced speech is closer to the reference speech as is the coded speech. This holds particularly for higher frequencies, which shows the efficacy of the proposed cepstral domain approach. Finally, we take a look on the cepstral coefficients in the bottom of Fig. 12. It can be seen that the cepstrum of the postprocessed speech is also closer to the reference speech, enhancing the cepstrum of coded speech not only into the same direction (e.g., $m = 2$ and $m = 4$), but even reversing the sign to better approach the reference cepstrum (e.g., $m = 10$ and $m = 24$).

Acknowledgment

The authors would like to thank S. Elshamy for providing an implementation of both DCT-II and the IDCT-II and J. Abel for advice concerning the setup of the subjective listening test.

References

[1] "Pulse code modulation (PCM) of voice frequencies," Telecommunication Standardization Sector (ITU-T), International Telecommunication Union, Geneva, Switzerland, Recommendation G.711, Nov. 1988.

[2] "Amendment 2: New Appendix III audio quality enhancement toolbox," Telecommunication Standardization Sector (ITU-T), International Telecommunication Union, Geneva, Switzerland, Recommendation G.711, Nov. 2009.

[3] "Wideband embedded extension for ITU-T G.711 pulse code modulation," Telecommunication Standardization Sector (ITU-T), International Telecommunication Union, Geneva, Switzerland, Recommendation G.711.1, Sep. 2012.

[4] J.-L. Garcia, C. Marro, and B. Kövesi, "A PCM coding noise reduction for ITU-T G.711.1," in *Proc. INTERSPEECH*, Brisbane, QLD, Australia, Sep. 2008, pp. 57–60.

[5] Y. Hiwasaki *et al.*, "G.711.1: A wideband extension to ITU-T G. 711," in *Proc. Eur. Signal Process. Conf.*, Lausanne, Switzerland, Aug. 2008, pp. 1–5.

[6] C. Plapous, C. Marro, L. Mauuary, and P. Scalart, "A two-step noise reduction technique," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Montreal, QC, Canada, May 2004, pp. 289–292.

[7] V. Ramamoorthy and N. Jayant, "Enhancement of ADPCM speech by adaptive postfiltering," *AT&T Bell Lab. Tech. J.*, vol. 63, no. 8, pp. 1465–1475, Oct. 1984.

[8] V. Ramamoorthy, N. Jayant, R. Cox, and M. Sondhi, "Enhancement of ADPCM speech coding with backward-adaptive algorithms for postfiltering and noise feedback," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 2, pp. 364–382, Feb. 1988.

[9] J.-H. Chen and A. Gersho, "Adaptive postfiltering for quality enhancement of coded speech," *IEEE Trans. Speech Audio Process.*, vol. 3, no. 1, pp. 59–71, Jan. 1995.

[10] T. Bäckström, *Speech Coding: Code Excited Linear Prediction*. New York, NY, USA: Springer, 2017.

[11] "Mandatory speech codec speech processing functions; adaptive multi-rate (AMR) speech codec; transcoding functions," 3GPP, Sophia Antipolis Cedex, France, 3GPP TS 26.090, Rel. 14, Mar. 2017.

[12] "Speech codec speech processing functions; adaptive multi-rate-wideband (AMR-WB) speech codec; transcoding functions," 3GPP, Sophia Antipolis Cedex, France, 3GPP TS 26.190, Rel. 14, Mar. 2017.

[13] "Codec for enhanced voice services (EVS); detailed algorithmic description," 3GPP, Sophia Antipolis Cedex, France, 3GPP TS 26.445, Rel. 14, Mar. 2017.

[14] R. Hagen, E. Ekudden, B. Johansson, and W. Kleijn, "Removal of sparse-excitation artifacts in CELP," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Seattle, WA, USA, May 1998, pp. 145–148.

[15] S. Han and T. Fingscheidt, "Improving scalar quantization for correlated processes using adaptive codebooks only at the receiver," in *Proc. Eur. Signal Process. Conf.*, Lisbon, Portugal, Sep. 2014, pp. 386–390.

[16] S. Han and T. Fingscheidt, "Lloyd-Max quantization of correlated processes: How to obtain gains by receiver-sided time-variant codebooks," in *Proc. 10th Int. ITG Conf. Syst., Commun. Coding*, Hamburg, Germany, Feb. 2015, pp. 1–6.

[17] Z. Zhao, S. Han, and T. Fingscheidt, "Improving vector quantization-based decoders for correlated processes in error-free transmission," in *Proc. 12th ITG Conf. Speech Commun.*, Paderborn, Germany, Oct. 2016, pp. 70–74.

[18] S. Han and T. Fingscheidt, "An improved ADPCM decoder by adaptively controlled quantization interval centroids," in *Proc. Eur. Signal Process. Conf.*, Nice, France, Sep. 2015, pp. 335–339.

[19] P. Vary and R. Martin, *Digital Speech Transmission: Enhancement, Coding and Error Concealment*. Hoboken, NJ, USA: Wiley, 2006.

[20] V. Grancharov, J. Plasberg, J. Samuelsson, and W. Kleijn, "Generalized postfilter for speech quality enhancement," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 16, no. 1, pp. 57–64, Dec. 2008.

[21] E. Jokinen, M. Takanen, M. Vainio, and P. Alku, "An adaptive post-filtering method producing an artificial Lombard-like effect for intelligibility enhancement of narrowband telephone speech," *Comput. Speech Lang.*, vol. 28, no. 2, pp. 619–628, Mar. 2014.

[22] G. Fuchs, A. Lombard, E. Ravelli, and M. Dietz, "A comfort noise addition post-processor for enhancing low bit-rate speech coding in noisy environments," in *Proc. IEEE Global Conf. Signal Inf.*, Orlando, FL, USA, Dec. 2015, pp. 498–502.

[23] Y. Wang and D. Wang, "Towards scaling up classification-based speech separation," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 21, no. 7, pp. 1381–1390, Mar. 2013.

[24] A. Narayanan and D. Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Vancouver, BC, Canada, May 2013, pp. 7092–7096.

[25] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Process. Lett.*, vol. 21, no. 1, pp. 65–68, Nov. 2013.

[26] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 23, no. 1, pp. 7–19, Oct. 2014.

[27] X. Lu, S. Matsuda, C. Hori, and H. Kashioka, "Speech restoration based on deep learning autoencoder with layer-wised pretraining," in *Proc. INTERSPEECH*, Portland, OR, USA, Sep. 2012, pp. 1504–1507.

[28] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder," in *Proc. INTERSPEECH*, Lyon, France, Aug. 2013, pp. 436–440.

[29] A. L. Maas, Q. Le, T. O'Neil, O. Vinyals, P. Nguyen, and Y. Andrew, "Recurrent neural networks for noise reduction in robust ASR," in *Proc. INTERSPEECH*, Portland, OR, USA, Sep. 2012, pp. 22–25.

[30] F. Weninger *et al.*, "Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR," in *Proc. Int. Conf. Latent Variable Anal. Signal Separation*, Liberec, Czech Republic, Aug. 2015, pp. 91–99.

[31] J. Lee, K. Kim, T. Shabestary, and H.-G. Kang, "Deep bi-directional long short-term memory based speech enhancement for wind noise reduction," in *Proc. Hands-Free Speech Commun. Microphone Arrays*, San Francisco, CA, USA, Mar. 2017, pp. 41–45.

[32] S. Park and J. Lee, "A fully convolutional neural network for speech enhancement," in *Proc. INTERSPEECH*, Stockholm, Sweden, Aug. 2017, pp. 1993–1997.

[33] S.-W. Fu, Y. Tsao, and X. Lu, "SNR-aware convolutional neural network modeling for speech enhancement," in *Proc. INTERSPEECH*, San Francisco, CA, USA, Sep. 2016, pp. 3768–3772.

[34] T. Kounovsky and J. Malek, "Single channel speech enhancement using convolutional neural network," in *Proc. Electron., Control, Meas., Signals Appl. Mechatronics*, Donostia-San Sebastian, Spain, May 2017, pp. 1–5.

[35] S.-W. Fu, T.-Y. Hu, Y. Tsao, and X. Lu, "Complex spectrogram enhancement by convolutional neural network with multi-metrics learning," in *Proc. Mach. Learn. Signal Process.*, Tokyo, Japan, Sep. 2017, pp. 1–6.

[36] S.-W. Fu, Y. Tsao, X. Lu, and H. Kawai, "Raw waveform-based speech enhancement by fully convolutional networks," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annual Summit Conf.* Kuala Lumpur, Malaysia, Dec. 2017, pp. 6–12.

[37] S.-W. Fu, T. Wang, Y. Tsao, X. Lu, and H. Kawai, "End-to-end waveform utterance enhancement for direct evaluation metrics optimization by fully convolutional neural networks," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 9, pp. 1570–1584, Sep. 2018.

[38] O. Abdel-Hamid, A.-R. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Kyoto, Japan, Mar. 2012, pp. 4277–4280.

[39] C. Dong, C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Feb. 2016.

[40] X.-J. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Proc. Adv. Neural Inf. Process. Syst.*, Barcelona, Spain, Dec. 2016, pp. 2802–2810.

[41] W. Shi *et al.*, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 1874–1883.

[42] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 1646–1654.

[43] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Santiago, Chile, Dec. 2015, pp. 1520–1528.

[44] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[46] Y. Ephraim and D. Malah, "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 6, pp. 1109–1121, Dec. 1984.

[47] X. Huang, A. Acero, H.-W. Hon, and R. Reddy, *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Upper Saddle River, NJ, USA: Prentice-Hall, 2001.

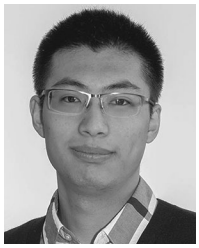
[48] "Multi-lingual speech database for telephonometry," NTT Advanced Technology Corporation, Kawasaki-shi, Japan, 1994.

[49] "Processing plan for G.711-plus (final version)," Telecommunication Standardization Sector (ITU-T), International Telecommunication Union, Geneva, Switzerland, Rapporteurs Q10/16, Oct. 2009.

[50] "Draft AMR-WB characterisation processing plan (WB-7c)," 3GPP, Erlangen, Germany, Version 1.0, Sep. 2001.

[51] A. Rämö and H. Toukomaa, "On comparing speech quality of various narrow- and wideband speech codecs," in *Proc. Int. Symp. Signal Process. Appl.*, Sydney, NSW, Australia, Aug. 2005, vol. 2, pp. 603–606.

[52] "EVS permanent document EVS-7c: Processing functions for characterization phase," 3GPP, Helsinki, Finland, v1.0.0, Aug. 2014.
[53] "Software tools for speech and audio coding standardization," Telecommunication Standardization Sector (ITU-T), International Telecommunication Union, Geneva, Switzerland, Recommendation G.191, Mar. 2010.
[54] "Objective measurement of active speech level," Telecommunication Standardization Sector (ITU-T), International Telecommunication Union, Geneva, Switzerland, Recommendation P.56, Dec. 2011.
[55] "40, 32, 24, 16 kbit/s adaptive differential pulse code modulation (ADPCM)," Telecommunication Standardization Sector (ITU-T), International Telecommunication Union, Geneva, Switzerland, Recommendation G.726, Aug. 1990.
[56] "7 kHz audio-coding within 64 kbit/s," Telecommunication Standardization Sector (ITU-T), International Telecommunication Union, Geneva, Switzerland, Recommendation G.722, Sep. 2012.
[57] "ANSI-C code for the adaptive multi-rate - wideband (AMR-WB) speech codec," 3GPP, Sophia Antipolis Cedex, France, 3GPP TS 26.173, Rel. 14, Apr. 2017.
[58] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, San Diego, CA, USA, May 2015, pp. 1–15.
[59] "Speech processing, transmission and quality aspects (STQ); speech quality performance in the presence of background noise; part 1: Background noise simulation technique and background noise database," ETSI, Sophia Antipolis, France, ETSI EG 202 396-1, Version 1.2.2, Sep. 2008.
[60] E. N. Gilbert, "Capacity of a burst-noise channel," *Bell Lab. Tech. J.*, vol. 39, no. 5, pp. 1253–1265, Sep. 1960.
[61] J. Abel, M. Kaniewska, C. Guillaume, W. Tirry, and T. Fingscheidt, "An instrumental quality measure for artificially bandwidth-extended speech signals," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 2, pp. 384–396, Feb. 2017.
[62] S. Elshamy, N. Madhu, W. Tirry, and T. Fingscheidt, "Instantaneous a priori SNR estimation by cepstral excitation manipulation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 25, no. 8, pp. 1592–1605, May 2017.
[63] "Perceptual evaluation of speech quality (PESQ)," Telecommunication Standardization Sector (ITU-T), International Telecommunication Union, Geneva, Switzerland, Recommendation P.862, Feb. 2001.
[64] "Mapping function for transforming P.862 raw result scores to MOS-LQO," Telecommunication Standardization Sector (ITU-T), International Telecommunication Union, Geneva, Switzerland, Recommendation P.862.1, Nov. 2003.
[65] "Wideband extension to recommendation P.862 for the assessment of wideband telephone networks and speech codecs," Telecommunication Standardization Sector (ITU-T), International Telecommunication Union, Geneva, Switzerland, Recommendation P.862.2, Nov. 2007.
[66] "Perceptual objective listening quality prediction (POLQA)," Telecommunication Standardization Sector (ITU-T), International Telecommunication Union, Geneva, Switzerland, Recommendation P.863, Mar. 2018.
[67] "Methods for subjective determination of transmission quality," Telecommunication Standardization Sector (ITU-T), International Telecommunication Union, Geneva, Switzerland, Recommendation P.800, Aug. 1996.
[68] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. Int. Conf. Mach. Learn.*, Long Beach, CA, USA, Jun. 2013, vol. 30, pp. 1–6.
[69] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter, "Self-normalizing neural networks," *in Proc. Adv. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 971–980.
[70] "Appendix I: A high quality low-complexity algorithm for packet loss concealment with G.711," Telecommunication Standardization Sector (ITU-T), International Telecommunication Union, Geneva, Switzerland, Recommendation G.711, Sep. 1999.
[71] "Adaptive multi-rate - wideband (AMR-WB) speech codec; error concealment of erroneous or lost frames," 3GPP, Sophia Antipolis Cedex, France, 3GPP TS 26.191, Rel. 14, Apr. 2017.
[72] D. Dean, S. Sridharan, R. Vogt, and M. Mason, "The QUT-NOISE-TIMIT corpus for the evaluation of voice activity detection algorithms," in *Proc. INTERSPEECH*, Chiba, Japan, Sep. 2010, pp. 3110–3113.
[73] Z. Zhao, H. Liu, and T. Fingscheidt, "Enhancements of G.711-coded speech providing quality higher than uncoded," in *Proc. 13th ITG Conf. Speech Commun.*, Oldenburg, Germany, Oct. 2018, pp. 211–215.
[74] J. W. Cooley and J. W. Tukey, "An algorithm for the machine calculation of complex Fourier series," *Math. Comput.*, vol. 19, no. 90, pp. 297–301, Apr. 1965.
[75] M. Narasimha and A. Peterson, "On the computation of the discrete cosine transform," *IEEE Trans. Commun.*, vol. COM-26, no. 6, pp. 934–936, Jun. 1978.



Ziyue Zhao received the B.Sc. degree in electronic and information engineering and M.Eng. degree in electronic and communication engineering from Xidian University, Xi'an, China, in 2012 and 2015, respectively. He is currently working toward the Ph.D. degree with the Institute for Communications Technology, Technische Universität Braunschweig, Braunschweig, Germany. His research interests include speech enhancement, improved speech and audio decoding, and deep learning approaches.

Huijun Liu received the B.Sc. degree in communication engineering from Xidian University, Xi'an, China, in 2014 and the M.Sc. degree in computer and communications systems engineering from Technische Universität Braunschweig, Braunschweig, Germany, in 2018. He is currently focusing on the application of deep learning in industrial automation.

Tim Fingscheidt (S'93–M'98–SM'04) received the Dipl.-Ing. degree in electrical engineering in 1993 and the Ph.D. degree in 1998 from RWTH Aachen University, Aachen, Germany. He further pursued his work on joint speech and channel coding as a Consultant with the Speech Processing Software and Technology Research Department, AT&T Labs, Florham Park, NJ, USA. In 1999, he joined the Signal Processing Department, Siemens AG (COM Mobile Devices), Munich, Germany, and contributed to speech codec standardization in ETSI, 3GPP, and ITU-T. In 2005, he joined Siemens Corporate Technology, Munich, Germany, leading the speech technology development activities in recognition, synthesis, and speaker verification. Since 2006, he has been a Full Professor with the Institute for Communications Technology, Technische Universität Braunschweig, Braunschweig, Germany. His research interests include speech and audio signal processing, enhancement, transmission, recognition, and instrumental quality measures. He is the recipient of several awards, which include a prize of the Vodafone Mobile Communications Foundation in 1999 and the 2002 prize of the Information Technology branch of the Association of German Electrical Engineers (VDE ITG). In 2017, he co-authored the ITG award-winning publication, but the ITG prize is only awarded once in a life time. He has been the speaker of the Speech Acoustics Committee ITG AT3 since 2015. From 2008 to 2010, he was an Associate Editor for the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING, and since 2011, he has been a member of the IEEE Speech and Language Processing Technical Committee.